A
Learning Project-I Report
On

"Smart Attendance System by using RFID "

Submitted in partial fulfillment of
The requirements for the 4th Semester Sessional Examination
of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING

By
Vivek Kumar
Sanjeev Kumar Mandal
Prabhat Kumar Mahato

Registration No.
21UG01LE60
21UG01LE56
21UG01LE52

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GIET UNIVERSITY, GUNUPUR

2022 - 23

GIET UNIVERSITY,GUNUPUR

Dist. - Rayagada, Odisha-765022, Contact:- +91 7735745535,
06857-250170,172, Visit us:- **www.giet.edu**

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the project work entitled "Smart Attendance System by using RFID" is done by Vivek Kumar , Sanjeev Kumar Mandal , Prabhat Kumar Mahato , Regd. No.- 21UG01LE60 , 21UG01LE56, 21UG01LE52 in partial fulfillment of the requirements for the 3rd Semester Sessional Examination of Bachelor of Technology in Computer Science and Engineering during the academic year 2022-23. This work is submitted to the department as a part of evaluation of 4th Semester Learning Project-II.

Proctors *Class Teacher*

Project Coordinator,2nd Year *HoD, CSE, 2nd year*

ACKNOWLEDGEMENT

We would like to thank our proctors **Mahesh Sir** and **Chinmaya Sir ,**

Secondly we would like to thank our class teacher **Dr. Bidush Kumar Sahaoo,** for guiding us well for the project.

And Thirdly we would like to thanks our Project Coordinator **Mr.Bhavani Sankar Panda** for well invigilation and great feedback about the project.

A special gratitude to our **HOD Dr. K.Murali Gopal and Dy. Dean, Computational Science Dr.Sanjay Kumar Kuanar,** for providing us this opportunity for making this project happen and and to show our development skills.

Thanks to all my team members for going strength to strength with me. Without them, this project is incomplete.

Vivek Kumar (21UG01LE60)
Sanjeev Kumar Mandal (21UG01LE56)
Prabhat Kumar Mahato (21UG01LE52)

ABSTRACT

Radio Frequency Identification (RFID) is a new technology in communication system which can be define as a medium used to identify and track the special tag implanted into an object or a living thing by using radio frequency wave. It is a wireless mean of communication that use electromagnetic and electrostatic coupling in radio frequency portion of the spectrum to communicate between reader and tag through a variety of modulation and encoding scheme. Nowadays, most of universities used the conventional method of taking attendance by calling names or signing on paper is very time consuming and inefficient. From that, by integrating various components which are RFID reader, RFID card, microcontroller and Secure Digital Card (SD Card), a portable RFID based attendance system can be set up and become the solutions to address this problem. Uniquely identify each person based on RFID tag is one of its special ability that can make the recording attendance process become more faster and easier compared to conventional method.

It consists of an RFID based attendance system that allows for automatic attendance marking by using RFID tags. Every student is provided with a unique authorization tag/card that is used to record his/her attendance.

A lot of time is wasted in schools and colleges for manual attendance procedures, in such cases our system provides an instant and automated attendance marking system. Every authorized student is provided with a unique RFID tag/card with his/her details fed in it. The tag consists of a built in integrated circuit that stores this data through modulating and demodulating transmitted radio frequency signals.

The data thus stored in this card is the unique identification of that person. As soon as the card is placed in front of the RFID reader, the data in it is read and attendance for that student is registered. This is done with the help of an 8266 microcontroller interfaced with the reader.
If it is a registered student, then buzzer will peak sound, else buzzer will not peak sound
.
This system will help the authority manage the attendance system in a more organized, efficient and time saving manner. The proposed method has been implemented in a prototype system that has proved the effectiveness of the system in easing the chores of attendance taking as a result of the automation of the system using the RFID technology. The design of the system is simple, inexpensive and portable making it a good candidate for commercial and academic purpose.

INTRODUCTION

In this system the concept of IOT is applied to attendance system of a classroom .

A portable module is designed which has the capability of recognizing the student via thier Id card , RFID reader reads the data through id card and then sending that data of that student to the server whose id card is recognized.

First of all the system requires connectivity to the internet, which can be achieved thorugh Wi-Fi. So the system is required which has the capability of wifi connectivity for which ESP 8266 is chosen.

Once the connenction is established, it scans for the Data via RFID tag(ID card) and recogonizes the student, whoever is recogonized their id card is sent the system .when the server receives the data it marks the attendence of the student.

PURPOSE :

An RFID-based attendance system makes use of RFID technology to mark student's attendance before every lecture in the institute. The students have to use an RFID tag to mark their presence on the RFID reader. The data is stored in the attendance system with high accuracy and efficiency.The microcontroller does the task of storing the attendance of the respective person in the Microcontroller memory.

PROJECT SCOPE :

Range of the RFID reader can increases ,so the reader can detect the tag from far distance.
RFID transreceiver module can be used for long distance communication to transfer data.
The transferred data can be save and store in computer as a database using specific software.
Timely attedance can be monitored i.e time in and time out can bar corded using real time clock and can be stored in the database.
This attendance system’s data base can be linked with college website and be shared and monitored by the student’s parents.
We can send data through internet to the user . So that user can access it remotely via internet.

PRODUCT FEATURES :

1. Reduce Faculty Workload :

With automation, the attendance process significantly reduces paperwork and saves the efforts of the educators of the teachers, which, in turn, augments productivity. Also, there is no need for roll-call for each lecture, thus saving the faculty additional time & effort of conducting manual attendance. Here, the teaching faculty can dedicate most of their time in teaching and improving the student learning process.

2. Saves Time :

The RFID takes attendance within a matter of a few seconds, unlike the conventional attendance process, just by displaying the card near the RFID reader. Moreover, the educators can access and authorize the attendance records within seconds from the database. With an easy student attendance record retrieval process, there is no need to maintain bulk records, thus saving a large number of hours.

3. 100% Accurate Attendance :

RFID attendance is carried out by providing each individual with an unique UIN number. When the card is scanned, the number is read and the attendance is marked, resulting in zero chance of human error. There is no scope of proxy attendance as each individual has a unique ID, resulting in the attendance process to be highly accurate.

4. RFID TAG IS REUSABLE :

If a student left the college then their RFID tag can be used for other students to store their data

WORK DONE IN AREA

1. COLLEGE:

The RFID cards/tag are then distributed to students and faculty whose attendance needs to be recorded. The student information is stored on a RFID tag. The tag transmits data as radio signals. The RIFD reader recognizes nearby RFID tag.

2. INDUSTRY:

It also can be used to take attendance for workers in working places. Its ability to uniquely identify each person based on their RFID tag type of ID card make the process of taking the attendance easier, faster and secure as compared to conventional method.

3. OFFICE:

RFID Attendance Management System (AMS) is monitoring the movements {In/Out) of Employee/Student /Staff/Visitor from one location to another location and provides visual representation on GUI and tracking process of Employee/Student /Staff/Visitor.

4. DOOR LOCKS:

One of the most popular use cases for RFID technology is guestroom entry. Compared to a traditional keycard, RFID-equipped cards offer hoteliers more control over security. Front desk staff can activate and deactivate cards remotely and review logs to see where and when a card was used. From a guest's perspective, RFID keycards are easier to use than traditional credit card-style keycards (simply wave it in front of the door lock transponder to open the door), and the ability to deactivate lost RFID key cards gives guests peace of mind. RFID cards can also be more cost-effective in the long term as they don't get demagnetized.

5. Attendance System for Teachers and Non-Teaching Staff :

While the RFID system is mainly used for tracking of students' attendance, the same system can be extended to track the attendance and movement of teachers and non-teaching staff without an increase in cost. The same hardware and software components can be used to generate reports on teachers and non-teaching staff attendance and behavior in the school campus.

SYSTEM ANALYSIS

HARDWARE REQUIRED :

1. ESP 8266 NODEMCU

In this project, we have used NodeMcu which is an opensource platform. NodeMcu is based on ESP8266 which can be used to connect objects and transfer data using Wifi. In this project, the NodeMCU is used to send the attendance on the spreadsheet via the wifi module. Whatever the google spreadsheet code has generated the deployment link, we have to copy that link and paste it into the code and with the help, nodeMCU will send the data on the spreadsheet.

2 . **RFID TAGS AND CARDS**

RFID TAG : The RFID tags are differentiated as passive and active tags. If the device doesn't have its power supply it is called a passive RFID tag. Thus, The passive tags have to be in very close range of an RFID reader and make use of the radio waves which are broadcasted by the reader to power the response alternatively if the device has its battery power to perform entire operations which are called active RFID tags.

RFID READER : An RFID reader consists of an antenna and a Radio frequency module which basically generates a high-frequency electromagnetic field. As we all know an RFID tag is a passive device which means that it doesn't have a power supply or a battery. An RFID reader has a microchip that is used to store and process the information. And also it has an antenna that is used to receive to transmit a signal. To read the information on the RFID tag it needs to be placed in very close range of the reader. An RFID reader basically generates an electromagnetic field that causes electrons to move through the tag in the antenna and it powers up the chip.

3.BUZZER :

This buzzer also has an important role in this module. Whenever the user will scan his/her card then this buzzer beeps for a few seconds so that the one who scanned the card will know that his card is scanned properly. Without a buzzer, one can only assume the card has been detected but he/she will not be sure so the buzzer is important here.

4.JUMPER WIRE :

Jumper wires are electrical wires with connector pins at each end. They are used to connect two points in a circuit without soldering. You can use jumper wires to modify a circuit or diagnose problems in a circuit.

SOFTWARE REQUIRED :

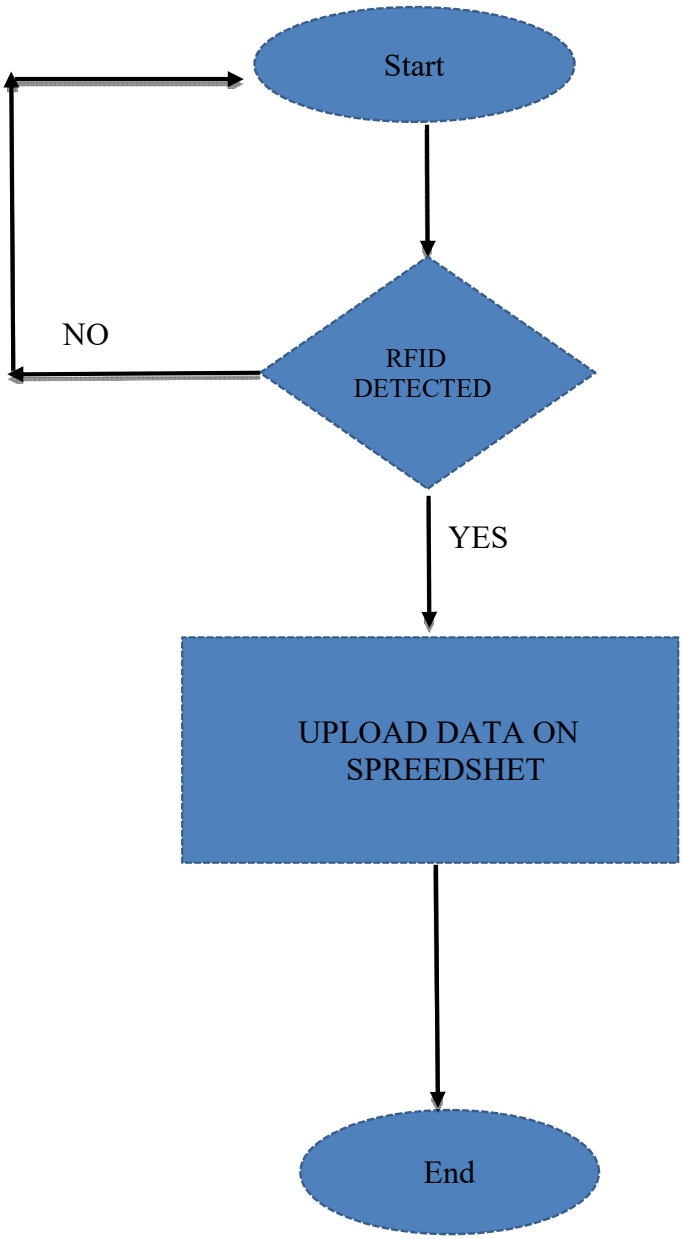
-- ARDUINO - IDE
-- GOOGLE SHEET DATABASE

LANGUAGE USED :

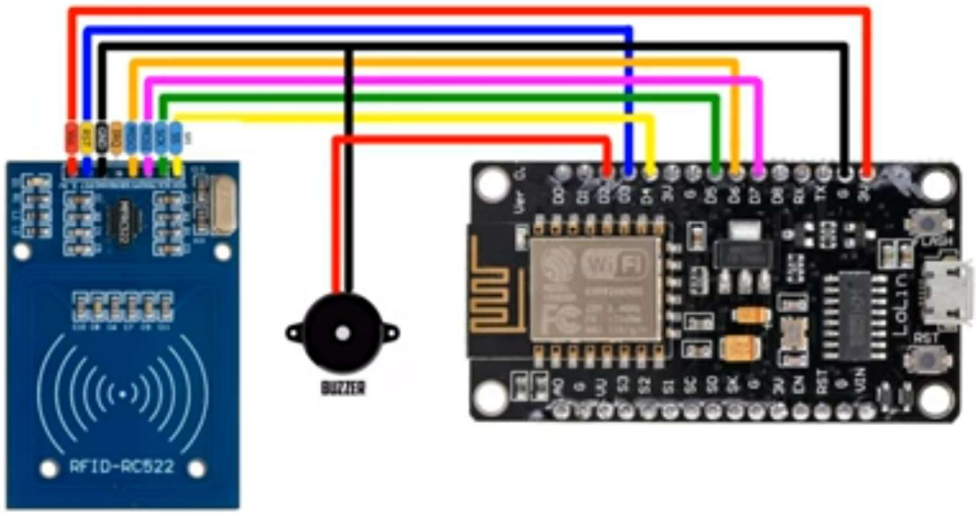
C++

SYSTEM DESING AND SPECIFICATION

FLOW CHART :



CONNECTION DIAGRAM :



RFID		**NODEMCU**
SDA	=>	D4
SCK	=>	D5
MOSI	=>	D7
MISO	=>	D6
GND	=>	G
RST	=>	D3
3.3V	=>	3V
BUZZER	=>	D2 and

SOURCE CODE

Cardholder_name code :

```
#include<SPI.h>
#include<MFRC522.h>
#include<ESP8266WiFi.h>
//---------------------------------------
constexpr uint8_t RST_PIN = D3;
constexpr uint8_t SS_PIN = D4;
//---------------------------------------
MFRC522 mfrc522(SS_PIN, RST_PIN);
MFRC522::MIFARE_Key key;
//---------------------------------------
/* write data */

int blockNum = 2;
byte blockData [16] = {"vivek_kumar"};

//---------------------------------------

byte bufferLen = 18;
byte readBlockData[18];
//---------------------------------------
MFRC522::StatusCode status;
//---------------------------------------



void setup()
{
  //---------------------------------------
  //Initialize serial communications
  Serial.begin(9600);
  //---------------------------------------
  //Initialize spi
  SPI.begin();
  //---------------------------------------
  //Initialize MFRC522 Module
  mfrc522.PCD_Init();
  Serial.println("Scan a MIFARE 1K Tag to write data...");
  //---------------------------------------
}
```

```
/// write data ////
void loop()
{

  for (byte i = 0; i < 6; i++){
    key.keyByte[i] = 0xFF;
  }
  //-------------------------------------------------------------------------------
  /* Looking for new cards */
  /* Reset the loop if no new card is present on RC522 Reader */
  if ( ! mfrc522.PICC_IsNewCardPresent()){return;}
  //-------------------------------------------------------------------------------
  /* Selecting  one card */

  if ( ! mfrc522.PICC_ReadCardSerial()) {return;}
  //-------------------------------------------------------------------------------
  Serial.print("\n");
  Serial.println("*Card Detected*");
  /* Print UID of the Card */
  Serial.print(F("Card UID:"));
  for (byte i = 0; i < mfrc522.uid.size; i++){
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
    Serial.print(mfrc522.uid.uidByte[i], HEX);
  }
  Serial.print("\n");
  /* Print type of card (for example, MIFARE 1K) */
  Serial.print(F("PICC type: "));
  MFRC522::PICC_Type piccType = mfrc522.PICC_GetType(mfrc522.uid.sak);
  Serial.println(mfrc522.PICC_GetTypeName(piccType));
  //-------------------------------------------------------------------------------
   /* Call 'WriteDataToBlock' function, which will write data to the block */
   Serial.print("\n");
   Serial.println("Writing to Data Block...");
   WriteDataToBlock(blockNum, blockData);
   //-------------------------------------------------------------------------------
   /* Read data from the same block */
   Serial.print("\n");
   Serial.println("Reading from Data Block...");
   ReadDataFromBlock(blockNum, readBlockData);
   /* If you want to print the full memory dump, uncomment the next line */
   //mfrc522.PICC_DumpToSerial(&(mfrc522.uid));
   //-------------------------------------------------------------------------------
   /* Print the data read from block */
   Serial.print("\n");
   Serial.print("Data in Block:");
   Serial.print(blockNum);
```

```
  Serial.print(" --> ");
  for (int j=0 ; j<16 ; j++){
    Serial.write(readBlockData[j]);
  }
  Serial.print("\n");
  //----------------------------------------------------------------------------
}



/**********************************
 * Writ() function
 **********************************/
void WriteDataToBlock(int blockNum, byte blockData[])
{
  //----------------------------------------------------------------------------
  /* Authenticating the desired data block for write access using Key A */
  status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, blockNum,
&key, &(mfrc522.uid));
  if (status != MFRC522::STATUS_OK){
    Serial.print("Authentication failed for Write: ");
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
  }
  //----------------------------------------------------------------------------
  else {
    Serial.println("Authentication success");
  }
  //----------------------------------------------------------------------------
  /* Write data to the block */
  status = mfrc522.MIFARE_Write(blockNum, blockData, 16);
  if (status != MFRC522::STATUS_OK) {
    Serial.print("Writing to Block failed: ");
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
  }
  else
  {Serial.println("Data was written into Block successfully");}
  //----------------------------------------------------------------------------
}

/////  ReadDataFromBlock  ///////////

void ReadDataFromBlock(int blockNum, byte readBlockData[])
{
  //----------------------------------------------------------------------------
```

```
  /* Authenticating the desired data block for Read access using Key A */
  status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, blockNum,
&key, &(mfrc522.uid));
  //-------------------------------------------------------------------------
  if (status != MFRC522::STATUS_OK){
   Serial.print("Authentication failed for Read: ");
   Serial.println(mfrc522.GetStatusCodeName(status));
   return;
  }
  else {
    Serial.println("Authentication successfully");
  }
  //-------------------------------------------------------------------------
  /* Reading data from the Block */
  status = mfrc522.MIFARE_Read(blockNum, readBlockData, &bufferLen);
  if (status != MFRC522::STATUS_OK){
    Serial.print("Reading failed: ");
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
  }
  else {
    Serial.println("Block was read successfully");
  }
}
```

ESTABLISH CONNECTION CODE :

```
#include <SPI.h>
#include <MFRC522.h>
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <WiFiClientSecureBearSSL.h>
//----------------------------------------
#define RST_PIN D3
#define SS_PIN D4
#define BUZZER D2
//----------------------------------------
MFRC522 mfrc522(SS_PIN, RST_PIN);
MFRC522::MIFARE_Key key;
MFRC522::StatusCode status;
//----------------------------------------
int blockNum = 2;
```

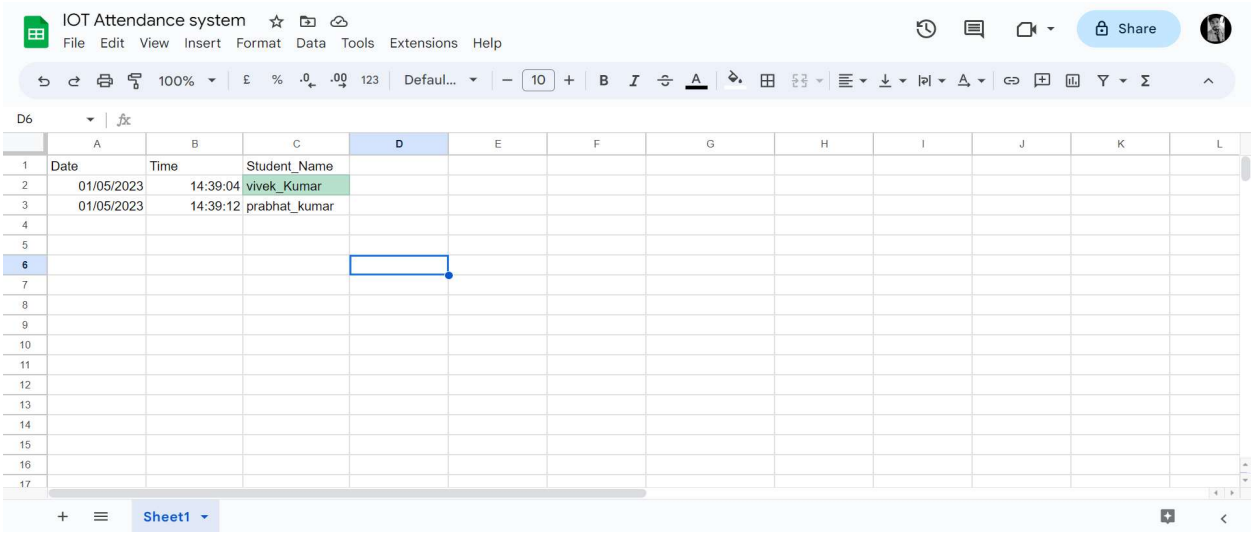
```
/* Creating another array to read data from Block */
byte bufferLen = 18;
byte readBlockData[18];
//----------------------------------------
String card_holder_name;
const String sheet_url=
"https://script.google.com/macros/s/AKfycbzzfOs8eWVs4_VJw8JJybvCSuJLphAnguIdjxnaIs0p
2Yd5JXwk6eZiPMJMJIc6qtVF/exec?name=";
//----------------------------------------
const uint8_t fingerprint[20] = {0xD5, 0x3A, 0x80, 0xA6, 0x03, 0xB0, 0xE4, 0x36, 0x0E, 0x46,
0x7B, 0x36, 0x45, 0xCB, 0x50, 0x4C, 0xD6, 0x98, 0xCE, 0x59};

//D5 3A 80 A6 03 B0 E4 36 0E 46 7B 36 45 CB 50 4C D6 98 CE 59
//----------------------------------------
#define WIFI_SSID "kattapa"
#define WIFI_PASSWORD "Ironman21"
//---------------------------------
//  setup() function

void setup()
{
  //-------------------------------------------------
  /* Initializing serial communications with the PC */
  Serial.begin(9600);
  //Serial.setDebugOutput(true);
  //-------------------------------------------------
  //WiFi Connectivity
  Serial.println();
  Serial.print("Connecting to AP");
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED){
    Serial.print(".");
    delay(200);
  }
  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  //-------------------------------------------------
  /* BUZZER as OUTPUT */
  pinMode(BUZZER, OUTPUT);
  //-------------------------------------------------
  /* Initialize SPI bus */
  SPI.begin();
  //-------------------------------------------------
```

```
}



///// loop() function

 void loop()
{
  //-------------------------------------------------
  /* Initializing MFRC522 Module */
  mfrc522.PCD_Init();
  /* Looking for new cards */
  /* Reset the loop if no new card is present on RC522 Reader */
  if ( ! mfrc522.PICC_IsNewCardPresent()) {return;}
  /* Select one of the cards */
  if ( ! mfrc522.PICC_ReadCardSerial()) {return;}
  /* Read data from the same block */
  //-------------------------------------------------
  Serial.println();
  Serial.println(F("Reading last data from RFID..."));
  ReadDataFromBlock(blockNum, readBlockData);
  /* If you want to print the full memory dump, uncomment the next line */
  //mfrc522.PICC_DumpToSerial(&(mfrc522.uid));

  /* Print the data read from block */
  Serial.println();
  Serial.print(F("Last data in RFID:"));
  Serial.print(blockNum);
  Serial.print(F(" --> "));
  for (int j=0 ; j<16 ; j++)
  {
    Serial.write(readBlockData[j]);
  }
  Serial.println();
  //-------------------------------------------------
  digitalWrite(BUZZER, HIGH);
  delay(200);
  digitalWrite(BUZZER, LOW);
  delay(200);
  digitalWrite(BUZZER, HIGH);
  delay(200);
  digitalWrite(BUZZER, LOW);
  //-------------------------------------------------
```

```
 if (WiFi.status() == WL_CONNECTED) {
  //-----------------------------------------------------------------------------
  std::unique_ptr<BearSSL::WiFiClientSecure>client(new BearSSL::WiFiClientSecure);
  //-----------------------------------------------------------------------------
  client->setFingerprint(fingerprint);
  // Or, if you want to ignore the SSL certificate
  //then use the following line instead:
  // client->setInsecure();
  //-----------------------------------------------------------------
  card_holder_name = sheet_url + String((char*)readBlockData);
  card_holder_name.trim();
  Serial.println(card_holder_name);
  //-----------------------------------------------------------------
  HTTPClient https;
  Serial.print(F("[HTTPS] begin...\n"));
  //-----------------------------------------------------------------


//NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
  if (https.begin(*client, (String)card_holder_name)){
   //-----------------------------------------------------------------
   // HTTP
   Serial.print(F("[HTTPS] GET...\n"));
   // start connection and send HTTP header
   int httpCode = https.GET();
   //-----------------------------------------------------------------
   // httpCode will be negative on error
   if (httpCode > 0) {
    // HTTP header has been send and Server response header has been handled
    Serial.printf("[HTTPS] GET... code: %d\n", httpCode);
    // file found at server
   }
   //-----------------------------------------------------------------
   else
   {Serial.printf("[HTTPS] GET... failed, error: %s\n", https.errorToString(httpCode).c_str());}
   //-----------------------------------------------------------------
   https.end();
   delay(1000);
  }
    else {
   Serial.printf("[HTTPS} Unable to connect\n");
  }

 }
 }
```

```
//// ReadDataFromBlock() function

void ReadDataFromBlock(int blockNum, byte readBlockData[])
{
  //-------------------------------------------------------------------------
  /* Prepare the ksy for authentication */
  /* All keys are set to FFFFFFFFFFFFh at chip delivery from the factory */
  for (byte i = 0; i < 6; i++) {
    key.keyByte[i] = 0xFF;
  }
  //-------------------------------------------------------------------------
  /* Authenticating the desired data block for Read access using Key A */
  status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, blockNum,
&key, &(mfrc522.uid));
  //-------------------------------------------------------------------------s
  if (status != MFRC522::STATUS_OK){
     Serial.print("Authentication failed for Read: ");
     Serial.println(mfrc522.GetStatusCodeName(status));
     return;
  }
  //-------------------------------------------------------------------------
  else {
    Serial.println("Authentication success");
  }
  //-------------------------------------------------------------------------
  /* Reading data from the Block */
  status = mfrc522.MIFARE_Read(blockNum, readBlockData, &bufferLen);
  if (status != MFRC522::STATUS_OK) {
    Serial.print("Reading failed: ");
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
  }
  //-------------------------------------------------------------------------
  else {

    Serial.println("Block was read successfully");
  }
  //-------------------------------------------------------------------------
}
```

TESTING/OUTPUT :



CONCLUSION :

In conclusion, the objective to build an RFID based attendance system was successfully achieved. In terms of performance and efficiency, this project has provided a convenient method of attendance marking compared to the traditional method of attendance system. A low cost RFID Based Attendance System prototype has been successfully developed. The prototype of the system can provide several benefits over the conventional method of taking attendance. This system will help in automatic storage of attendance in the database..

LIMITATIONS :

> **Limited Storage:**

Even most non-volatile memory requires power to keep its contents from degrading. Without an onboard battery, passive tags can't store much information. Requires Reader

> **Short Range -**

The absence of an active antenna means that a passive tag has an extremely short frequency range. Tags operating in low to high frequency may have a maximum communication range of 2 feet. If they’re using a very high frequency they may get up to 20 feet, but that’s still a relatively short range.